Adaptive and Convex Optimization-Inspired Workflow Scheduling for Cloud Environment

Kamlesh Lakhwani, Department of Computer Science and Engineering, JECRC University, Jaipur, India
https://orcid.org/0000-0002-4731-5179

Gajanand Sharma, Department of Computer Science and Engineering, JECRC University, Jaipur, India
https://orcid.org/0000-0001-6412-0392

Ramandeep Sandhu, School of Computer Science Engineering, Lovely Professional University (LPU), Jalandhar, India

Naresh Kumar Nagwani, Department of Computer Science and Engineering, National Institute of Technology, Raipur, India

Sandeep Bhargava, EHR Logic IT Services Pvt. Ltd., India

Varsha Arya, Department of Business Administration, Asia University, Taiwan & Lebanese American University, Beirut, Lebanon & Chandigarh University, Chandigarh, India*

Ammar Almomani, School of Computing, Skyline University College, Sharjah, UAE & Al- Balqa Applied University, Jordan
https://orcid.org/0000-0002-8808-6114

ABSTRACT

Scheduling large-scale and resource-intensive workflows in cloud infrastructure is one of the main challenges for cloud service providers (CSPs). Cloud infrastructure is more efficient when virtual machines and other resources work up to their full potential. The main factor that influences the quality of cloud services is the distribution of workflow on virtual machines (VMs). Scheduling tasks to VMs depends on the type of workflow and mechanism of resource allocation. Scientific workflows include large-scale data transfer and consume intensive resources of cloud infrastructures. Therefore, scheduling of tasks from scientific workflows on VMs requires efficient and optimized workflow scheduling techniques. This paper proposes an optimised workflow scheduling approach that aims to improve the utilization of cloud resources without increasing execution time and execution cost.

KEYWORDS

Bayesian Optimization, Cloud Optimization, Scientific-Workflow, Tabu Search, Whale Optimization, Workflow Scheduling

INTRODUCTION

Cloud is a challenging and highly demanding system where services are metered, reliable, and can be accessed on-demand Yang and Chen (2010), Zhang et al. (2010), Sandhu and Lakhwani (2022), Sorkhoh et al (2020). Workflows Zhao et al. (2011) have been used to model scientific applications Barker and Van Hemert (2007), Pietri et al. (2013), Gil et al. (2007). Scientific workflows like

DOI: 10.4018/IJCAC.324809
*Corresponding Author

MONTAGE, LIGO, SIPHT, GENOME, etc. have millions of tasks. Zhao et al. (2011), Vöckler et al. (2011), Kouatli, I (2020). These tasks must be mapped to cloud resources as they become feasible to provide efficient scheduling with the least amount of resource consumption. There are varieties of optimization approaches that may be used to find optimal scheduling solutions (Casavant and Kuhl, 1988). To address different optimization problems numerous general-purpose meta-heuristic algorithms are available (Talbi, 2009). These algorithms provide scheduling and optimization solutions that are close to optimum (Kumar and Sivakumar, 2022; Bisht and Vampugani, 2022; Alakbarov, 2022), Farhat et al (2020) .

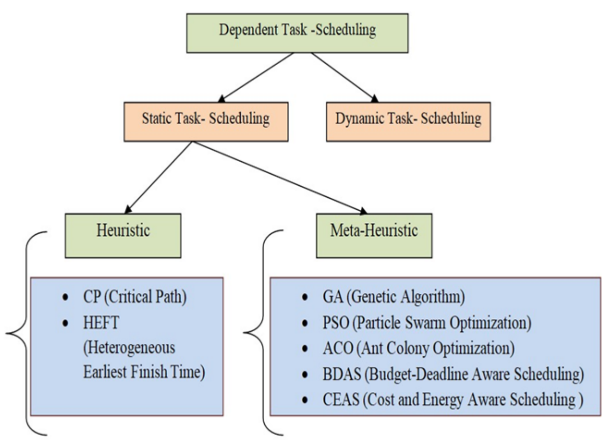
Meta-heuristic approaches are generally more computationally intensive than heuristic approaches and take longer to run; however, they also tend to find more desirable schedules as they explore different solutions using a guided search. In cloud systems, using meta-heuristics approach to solve the workflow scheduling problem involves many challenges such as: modeling a theoretically unbound number of resources, defining operations to avoid exploring invalid solutions (e.g., data dependency violations) to facilitate convergence, and pruning the search space by using heuristics based on the cloud resource model (Negi et al., 2013; Rajput et al., 2022; Kumar et al., 2022).

In the recent era, single-objective and multi-objective-based tasks scheduling (Vöckler et al., 2011, Vecchiola et al., 2009, Malawski et al., 2015, Hammoud et al. 2020) and mapping algorithms have been used by researchers in cloud environment (Holland, 1992, Rodriguez and Buyya, 2017, Mehta et al., 2009, Deelman et al., 2015, Verma and Kaushal, 2015). There are many promising studies that provide efficient scheduling of input tasks in cloud systems (Yu and Buyya, 2005, Calheiros et al., 2011, Arisdakessian,et al. 2020, Mishra et al. 2021). Still, research demands improvements in existing meta heuristic algorithms so that resources can be utilized at maximum.

Tasks scheduling in cloud computing is a fast and demanding area of research. In cloud computing, plenty of tasks runs concurrently and use the resources online. The scheduling reduces the computation time and processing time of tasks (Alkhanak et al., 2016, Liu et al., 2017, Reddy and Kumar, 2017, Rimal and Maier, 2016, Zhu et al., 2015, Wahab, O. A., et al. 2017). Different types of algorithms and techniques (Nasr et al., 2014, Singh and Singh, 2013, Zhang et al., 2017, Priya and Kiranbir, 2018, Huang et al., 2013, Abrishami et al., 2013, Arabnejad et al., 2016, Ghose et al., 2017, Shaw and Singh, 2014, Shi and Dongarra, 2006, Delavar and Aryan, 2014, Alkhanak and Lee, 2018, Al-Qerem et al. 2020, Kuppusamy, P., et al. 2022) used for task scheduling in the cloud system are categorized in Figure 1.

Figure 1 depicts methods of task scheduling where meta-heuristic approach is stronger and is adopted in proposed study.

Figure 1. Dependent task scheduling

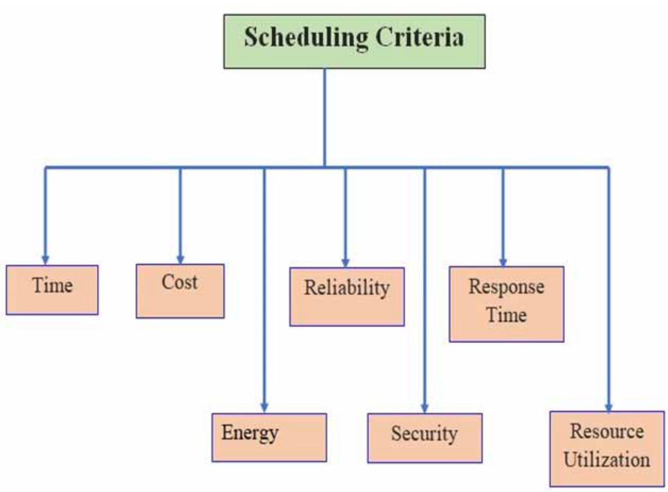


In scientific workflows and their mapping techniques, researchers have targeted various parameters to attain better outcomes. Figure 2 highlights these important scheduling parameters. Time From the age of grid computing, time or makespan is the primary objective of most scheduling techniques. As far as the Cloud Computing environment is concerned, execution time plays a significant role since the cloud provider charges its customers based on execution time. In workflow applications, the execution time is defined as the time taken to complete all tasks in a workflow. Hence reducing the execution time of workflow applications become a crucial factor. Cost Cloud providers often charge clients for leasing infrastructure, which includes costs for resource consumption, data transport, and cloud storage, among other things. The computation cost plays a dominant role in workflow scheduling, it is necessary to minimize these costs for the effective usage of the cloud platform. Thus, an efficient scheduling algorithm that considers these costs during the resource provisioning is necessary for executing the workflow applications in the heterogeneous cloud environment.

Energy Due to the rising execution of workflows in various fields, the energy consumption of data centers has gradually increased. Hence, energy conservation in cloud data centers has become a matter of concern. High energy consumption incurs high operational and maintenance costs. Due to an inefficient scheduling approach, a data center with a low workload may become a high energy consumption center. In addition, if the resources on the servers are over-utilized, the cloud system is classified as inefficient in terms of energy usage. Thus an effective scheduling mechanism is required to address these issues that will help to attain a green-environment by reducing unnecessary power consumption (Sharma and Sajid, 2021; Gupta and Gupta et al., 2018; Bhushan and Gupta, 2017, Gaurav et al. 2022, Zhang et al. 2017, Shahzad et al. 2022).

Resource utilization Resource utilization determines the efficient usage of resources. Leased Resources should be utilized efficiently to avoid unnecessary money expenditure as providers also charge for the unutilized slots. Improving resource utilization has considerable benefits for its various users in the form of cost and also for its providers in terms of profit and energy consumption. Hence improving resource utilization becomes a significant factor in scheduling. This research work focuses on the significant scheduling criteria related to economic factors such as Time, Cost, and environmental factors, such as consumption of energy and Utilization of the resources for computing Workflow applications in the cloud environment. Security Data privacy and security need to be addressed while adopting cloud computing as the workflows may contain confidential information which scientists

Figure 2. Scheduling parameters



may not wish to reveal (Alkhanak et al., 2016, Gaurav et al. 2022, Mehla R & Psannis, 2022, Kumar et al., 2022, Gupta et al., 2009, Quamara et al., 2019).

Cloud Computing is a world where internet-based computing exists. So, all services, whether they are storage, apps, or servers, are delivered to users' computers through the internet. For the success of the scientific world, the cloud has taken a well-built stride toward the facility of virtualization. There have been a lot of advancements made in this area. The primary objective of various workflow scheduling algorithms is to minimize the cost of execution. A majority of algorithms take other metrics, as outlined in Figure 3.

While Scheduling occurs, the important objectives are about what to minimize and what to maximize in the whole process (Vecchiola et al., 2009, Shaw and Singh, 2014, Singh, A., and Kumar, R 2021, Bhardwaj et al. 2022, Liang, Y., et al 2022). As depicted in Figure 3, make-span or total execution time (TET), total execution cost(TEC), total energy consumption, and response time should be minimized. On the other hand, the utilization of cloud resources should be maximized.

Major Contribution

The proposed method TBW contributes toward mapping and migration of tasks under deadline constraints. It also minimizes time and cost parameters. TBW method achieves better results than other optimization algorithms in the cloud VM scheduling. The objectives of this research are to optimize TEC, TET, and response time while migrating tasks from less utilized VMs to other VMs. Through simulation, it was observed that the proposed TBW algorithm outperforms the other optimization approaches.

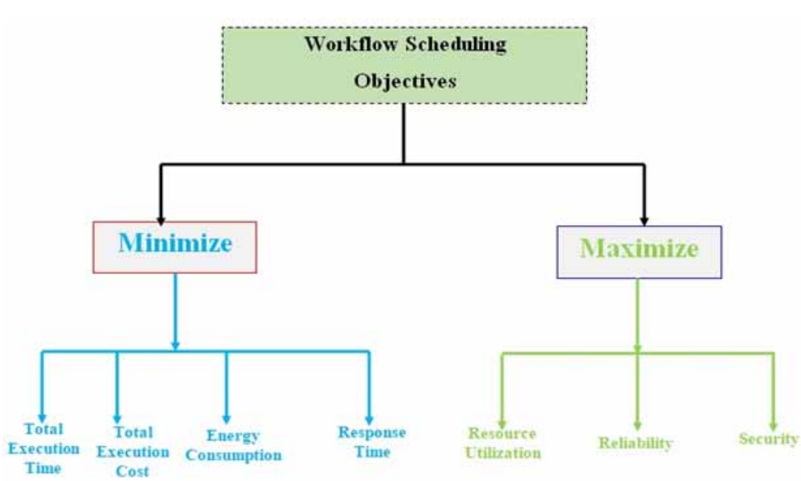
Paper Organization

The rest of the article is structured as follows. Section 2 presents some of the important literature work related to task ranking and resource optimization. Then the architecture and framework of the proposed methodology are presented in section 3. Experimental simulation, results, and performance analysis are represented in section 4. Section 5 provides the comparative analysis of the outcome with other state of art literature. A few significant directions for future study are outlined in Section 6.

RELATED WORK

Scheduling based problems need some meta-heuristic algorithms (Rodriguez and Buyya, 2014, Arabnejad and Barbosa, 2014, Ghose et al., 2017, Jiang et al., 2017, Bisht et al. 2022, Onyebuchi, et

Figure 3. Objectives behind scheduling a workflow



al. 2022). Such algorithms work on parameters like time, cost, response time, memory consumption, energy consumption, etc. In Karaboga and Basturk (2007), a hybrid tabu and ABC scheduling approach for cloud systems have been defined. The target of the study is to balance the load on VMs in the cloud. Further, based on speedup, utilization, total time, efficiency, energy consumption, and makespan parameters a comparison with existing scheduling methods has been done in this paper. Byun et al. (2011) created an algorithm that calculates the ideal amount of resources that should be leased to keep the cost of a workflow's execution to a minimum. The algorithm is made to run online and also creates a job for resource mapping. Every billing period (i.e., every hour), the schedule and resources are changed based on the state of the active VMs and tasks. In GA-PSO, optimal workflow scheduling results have been generated with load balancing; the authors used a hybrid approach for workflow tasks scheduling. Also, the researchers have done comparisons with existing methods like GA, PSO, WSGA, and HSGA (Al-Maamari and Omara, 2015).

In WOA Mirjalili and Lewis (2016), the work is on three operators. Searching for prey, trap the prey and then bubble-net foraging behavior of whales is the main dedication of the authors. Also, a comparison has been done in WOA on 26 mathematical benchmark functions. The optimization-based results have been compared with existing optimization algorithms like HGA, PSO, PSOPC, SOS, HPSO, MBA for different design problems. WOA has provided better results.

Sossa (2016) suggested a PSO-based approach to reduce the execution cost of a single process while balancing task load on available resources. While cost reduction is a top priority in clouds, load balancing is a more sensible goal in a non-elastic setting like a cluster or a grid. The workflow execution time is not addressed in the scheduling objectives; hence this number might be quite high as a result of the cost-cutting philosophy. The authors assume a certain number of VMs are available and ignore the cloud's flexibility. Because of this, the proposed solution is comparable to those used for grids, where the generated schedule is a mapping between tasks and resources rather than a more detailed schedule indicating the quantity and type of resources that need to be leased, when they should be acquired and released, and in what order the tasks should be carried out on them (Elrotub et al., 2021; Singh and Kumar, 2021; Xu et al.,2021).

In paper Liu et al. (2017), the authors have worked on a deadline constraint-based workflow scheduling process. They accepted four scientific workflows as Cybershake, Montage, Lego, and Inspiral. Within the user's defined deadline constraints, Both TET and TEC were evaluated. Apart from this, focus on crossover and mutation probability was also a prior concern. Performance is evaluated by a task ranking system. DAG is used to represent workflows. A penalty function, as well as a penalty rule in CGA was proposed which is CGA2 and it works without any parameter. Also, it has worked to overcome prematurity.

Authors in paper Reddy and Kumar (2017) have discussed the whale optimization-based algorithm which mimics the humpback whales. The authors shared that these whales hunt their food which is many small fishes close to the surface. For this hunting, whales swim around them within a shrinking circle.

Dubey et al. (2018) have provided the proposal for task ranking. In this scheduling, the rank of tasks and the assignment of processors are two important factors. Firstly, it creates a DAG and then it works on tasks in order. The proposed algorithm uses a modified HEFT algorithm that has reduced makespan and provides better resource utilization.

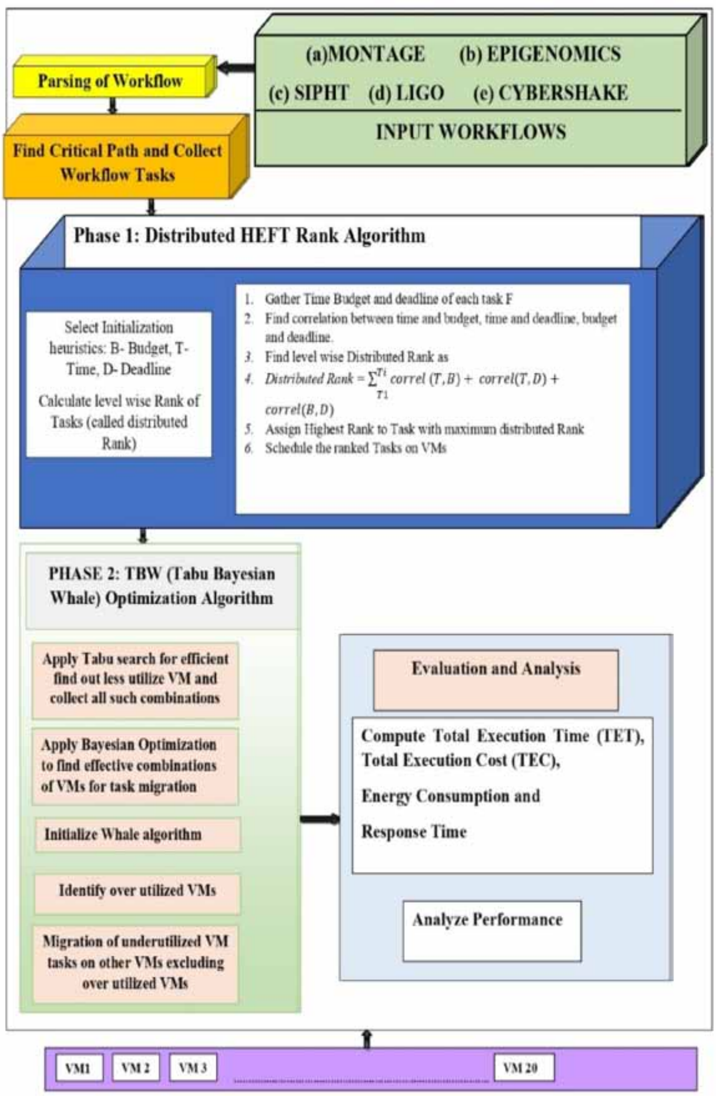
Alkhanak and Lee (2018) proposed a cost optimization approach for scientific workflow scheduling in cloud computing. The proposed method employs the four meta-heuristic algorithms and, these algorithms work on the VMs population of a cloud system. It helps in reducing the cost and time of the service providers. The execution cost and time are reduced as compared to baseline approaches.

Choudhary et al. (2018) introduced a gravitational search algorithm for workflow scheduling in the cloud environment. The optimizations in workflow reduce the cost and makespan. In this process, two algorithms are hybridized GSA and HEFT for workflow scheduling. The performance evaluation is done on the basis of two metrics that are monetary cost ratio and schedule length ratio.

PROPOSED METHODOLOGY

According to (Rodriguez and Buyya, 2014, Arabnejad and Barbosa, 2014, Vecchiola et al., 2009) in workflow scheduling, mapping of task on VMs should not be static. The proposed methodology is more effective in mapping the task dynamically. It works on two objectives. First, it targets effective task distribution on cloud resources, and second, it targets optimal scheduling for better performance. Figure 4 presents the architectural framework of proposed workflow scheduling and optimization technique. It consists of two phases. Workflow task ranking, using Distributed HEFT technique is performed in phase-1 and, resource optimization using proposed TBW method is performed in phase

Figure 4. Scientific workflow scheduling framework



2. Collection of the scientific workflow tasks are the input for phase-1. Ranked-tasks mapped on cloud VMs is the outcome of phase-1 which further optimized dynamically in phase-2 to improve TET and TEC for input workflow.

Methodology steps for complete framework are as follows:

1. Input the workflows.
2. Parse the tasks.
3. Ranking of tasks.
4. Provides the virtual machines according to ranking based paths.
5. Initialize the optimization using tabu search and Bayesian optimization.
6. Use whale optimization and update the status of the fitness function.
7. Check the output is optimized or not. If yes then analyze otherwise again initialize.
8. Analyze the total resource utilization.

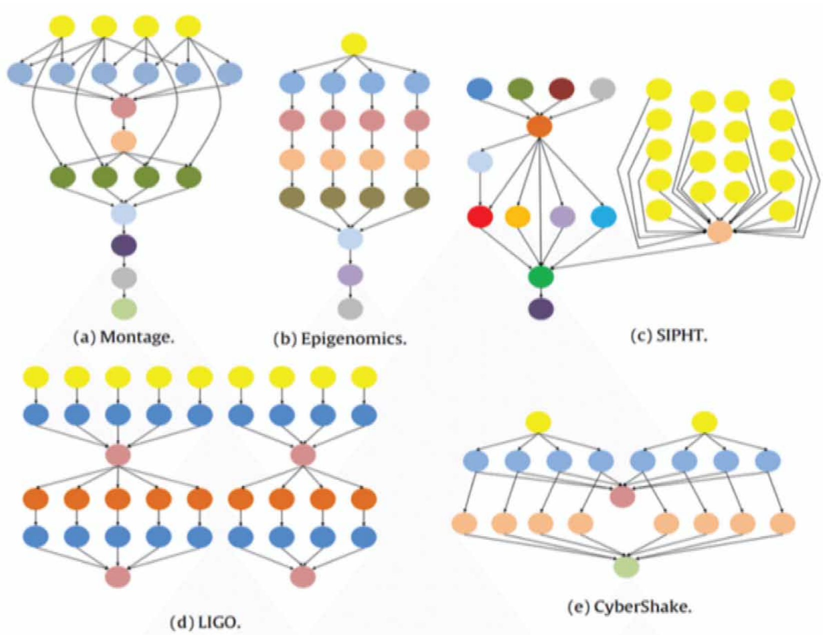
Input Workflow

Total five workflows are accepted as input named:

1. MONTAGE
2. CYBERSHAKE
3. SIPHT
4. LIGO
5. EPIGENOMICS

Graphical representation of these workflow is shown in Figure 5. These are important scientific applications and uses datasets at large scale.

Figure 5. Scientific workflows



Parsing and Finding Critical Path

Parsing is the next step. It is a step of analyzing input. In the proposed cloud workflow framework, parsing occurs at the initial stage. It shows tasks with dependencies. Following the critical path, all tasks of the input workflow were collected and analyzed. After this, phase 1 of the system begins.

Phase 1: Task Ranking

This phase progresses to find out the effective rank value of each task of input workflow. Distributed HEFT task ranking algorithm is proposed in this phase and various steps of this technique are represented in Algorithm 1.

Algorithm 1: Distributed HEFT task ranking

1: Begin
2: Select initialization heuristics:
3: B as Budget, T as Time, D as Deadline
4: Gather T, B, and D of each task F 5: Find Correlations as:
6: C_{TB} = correlation between T and B
7: C_{TD} = correlation between T and D 8: C_{BD} = correlation between B and D
9: For every task at each level compute:
10: Distributed score = $\sum\left(C_{TB}+C_{TD}+C_{BD}\right)$
11: Assign the highest rank to a task with a maximum correlation score
12: Schedule the Ranked task to VMs

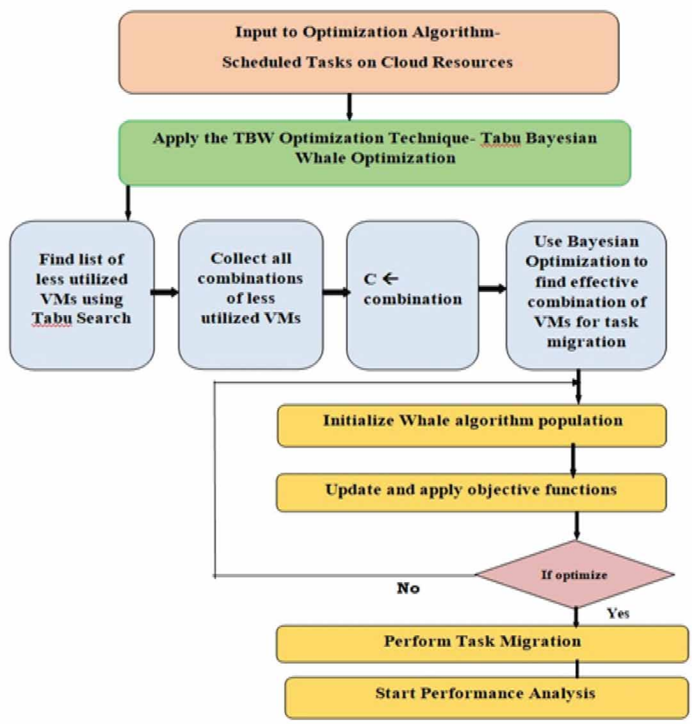
It works on three heuristic parameters budget, time, and deadline. The distributed HEFT ranking method finds out the correlation between these parameters and then assigns the top rank to the task having a highly distributed score among all tasks. A correlation's value might fall between -1 and +1. It is a relationship between two parameters that conveys how closely related two parameters are to one another. The task with the highest correlation receives the highest rank, while the task with the lowest correlation receives the lowest rank. Now, utilising the distributed-HEFT Rank, these rated jobs are scheduled on cloud resources.

Phase 2: Cloud Optimization

This phase further uses three optimization approaches to make the system more efficient. In order to reduce energy usage, it first executes task migration on any underused machines that may be present in the system. An innovative optimization strategy based on Tabu search, Bayesian, and whale optimization methodologies has been applied throughout this procedure. Tabu optimization helps to find out underutilized resources. Afterward, Bayesian optimization approach helps to provide a combination of VMs which are best suitable for task migration. While minimising time, cost, and reaction time, Whale optimization facilitates in the transfer of jobs from underutilized machine to others host. The major goal of this study also includes analysis and assessment of performance. The Makespan, Cost, Energy Consumption, and Response-time of Workflow Execution are determined after conducting VM migration in an efficient way. Figure 6 represents complete process of proposed TBW optimization technique. Moreover, an explanation of the proposed work with the algorithm and a flow chart of the methodology in detail is explained in this section. The concept of scheduling for optimization is implemented on the cloud system. Cloud-based virtual machines (VMs) are the most advantageous elements of this phase (Nigam et al., 2022; Kumar et al., 2022; Hemrajani et al., 2022, Stergiou et al. 2021, Gupta et al. 2021).

The Cloud-based system can be optimized utilizing Tabu, Bayesian, and Whale optimization approaches, as depicted in figure 6. It takes input tasks that have been ranked and assigned to cloud VMs using the distributed HEFT ranking algorithm. TBW algorithm is denoted as an optimization

Figure 6. TBW optimization technique in cloud environment



algorithm as it works step by step to find an optimum solution based on objective functions. All these steps are part of the schedule to ensure that neither any Virtual Machine is left idle for neither long periods of time nor any Virtual Machine is overloaded with work. A scheduler decides which task/job should go to which machine. Looking for underused VMs and searching for over-utilized VMs are the two primary processes done throughout the setup. TBW algorithm is capable of avoiding local optima. So, it is suitable for most practical applications. Apart from this, to solve different constrained or unconstrained optimization problems, no alteration is required to perform in the algorithm. In the proposed ideology, three algorithms are used: -Tabu Search to find the underutilized resources, Bayesian Optimization to combine the best suitable VMs for task migration, and Whale Optimization to optimized the task migration.

- **Tabu Search:** Input to Tabu search method is the list of VMs on which ranked-tasks are mapped. Tabu search algorithm is applied to find neighbours of current VM. It works iteratively. If a neighbour-VM is less utilized than the current-VM, add such neighbour in the Tabu-list. The outcome of the Tabu search is a final list of VMs that aren't being used effectively. Now, target the tasks of these VMs and migrate on other VMs (Efficiently utilized) but without making Queue. Algorithm 2 presents various steps incurred in determination of Tabu list for Tabu search in real time.
- **Bayesian Optimization (BO):** BO is a viable approach for determining which VM is most appropriate for a given workload. In this research work, it is used to choose the best combination of VMs for mapping tasks. It targets tasks of the VMs which are not efficiently utilized and those VMs which are in the underutilized category. BO provides all combinations of VMs where

tasks can be shifted. Algorithm 3 presents various steps incurred in the determination of the best combinations of VMs for mapping the task in real time.

- **Use of Whale Optimization (WO):** Whales have the ability to locate their target before engulfing them. Whale optimization algorithm is inspired by humpback whales' bubble-net feeding technique, in which they release a stream of bubbles in decreasing circle and spiral patterns around their pray. The whale positions are chosen at random and further analysed to see which the best is now. After then, the other whales also change their locations in accordance with the current optimal solution as described in Algorithm 4.

In Algorithm 4, p is a random number between 0 and 1. In step-6, $X(t+1)$ is the updated position, $X¢(t)$ is the best solution and method $A = (2*\alpha*r) - \alpha$ where $\alpha$is linearly decreased from 2 to 0 depends on max iteration number shrinking encircling and r is random vector between (0, 1) and $D = | (2*r*X¢(t))-(X(t))|$where $X(t)$ is position vector. In step-7, $D = |(2*r*X_{rand}(t)) - (X(t))|$and $X_{rand}(t)$ is position vector. In step-8, b defines the logarithmic spiral shape, L is random number between −1 and 1 and $D' = X¢(t) - X(t)$.

In this research work, optimization is used for the migration of tasks from underutilized machines to other ones but ones, without any increase in time, cost and response time. Whale Optimization takes input from Bayesian Optimization. Based on its objective functions, it chooses the best combination of VM and shifts tasks on VMs efficiently without increasing TEC and TET. Overall results are better than GA-PSO for scientific workflows. In the whole process, management of time and cost are the main factors.

Equations (1 to 5) expressed below are effectively elaborating TET and TEC used throughout the process:

$$T_{t} = \text{å}T_{R} + \text{å}T_{P} + \text{å}T_{W} \tag{1}$$

where:

T_t : Total Time
T_R: Receiving or passing time of task
T_P: Processing Time of task
T_W: Waiting Time of task

$$\text{Actual Cost} = \text{Under Deadline Total Cost} + \text{Deadline crossed Task Cost} \tag{2}$$

$$\text{Total Cost} = (MF + CF)/2 \tag{3}$$

where:

MF: Movement Factor
CF: Cost Factor

$$\text{MF} = \text{Number of Migrations/used VMs} \tag{4}$$

$$\text{CF} = (\text{Process Cost} * \text{task memory})/\text{involved VMs} \tag{5}$$

Algorithm-5 named TBW optimization algorithm has used algorithm-2 Tabu search to begin the process. It uses a tabu list which stores list of those VMs which are not effectively utilized. Then on the basis of tabu list, statement 6 in TBW algorithm is used to call Bayesian optimization algorithm which is algorithm 3. Bayesian optimization provides best combination of resources, as the target of this research is to migrate tasks from underutilized VMs to other VMs. Using it, algorithm 4 has started from statement 8 which is use of whale optimization for actual migrate of tasks of underutilized machines to other VMs. Also the complete process has analyzed the scheduling on the basis of time, cost, and response time and energy consumption parameters.

Algorithm 2: Tabu Search

```
1:  procedure TABUSEARCH (VMs)
2:  Begin:
3:  X^best := X^now := randomly choose VM from VMs list
4:  for each VM do
5:     G1: ={x|neighbour of X^now and x satisfies the rule}
6:     G2:={x|neighbour of X^now and x does not satisfy the rule}
7:  for x in G1 do
8:     X^now:= x that minimizes f(x) → where f(x) utilization
       function for VM x
9:     if f(x) < fX^best) and fx) < fX^nw) for some x in G2 then X^now :=x
10:    if f(X^now) < f(X^best) then X^best:= X^now
11:       Tabulist:=X^best and f(X^best)
12: Return Tabulist
```

Algorithm 3: Bayesian Optimization

```
1: Procedure BAYESOPT(VMs,Tc) Begin:
2: Initial combination XΘ¬ Tc
3: Bound of length Θ ∈X_L,X_U
4: θ_P∈{ X_L, X_U, P}
5: Workflow size:K
6: for t = 1 to task_i do
7: X_i: n = argmax(αt(X|Pr(XL−i,θ: n))
8: X_final= Evaluate(X_L−I, X_i: n)
Return X_final
```

Algorithm 4: Whale Optimization

```
1: Procedure WHALE(population)
2: Begin
3: while t < t_max do
4: for each agent do
5: if p <0.5 where p is the random number between 0 and 1 then
6: if |A|<1 then X(t+1) = X'(t) - A·D^n
7: else if |A| >= 1 then X(t+1) = X_rand(t) - A·D^n
8: else if p >= 0.5 then X(t+1) = D'·e^bi cos (2πl)+ X'(t)
Evaluate the fitness of X(t + 1) and updates X′
```

Algorithm 5: TBW Optimization Algorithm

```
procedure TBW()
Begin
Take input as ranked scheduled Tasks on VMs
VMs = TABU (task, rank)        ▷ call algorithm 2
Tc ← Task all combination
Xfinal= Bayesopt(VMs, Tc) ▷ call algorithm 3
for each X_final(i) do
   Whale(i)← X_final(i) ▷ whale optimization used
Update whale parameters a, A, C and L
      Calculate distance Between each whale(i)
      D= C. X_final−X_rand
      if A < 1 then
      Update current whale position
X(t+1)=1/data centre 1/VM (Memory cost + processor cost + time delay)
      else
      X(t)← threshold
      Th← threshold
      According to Th migrate task to VM
      Analysis and Performance Evaluation
Analyse the parameters total execution cost (TEC), total
execution time (TET), response time (RT) and energy consumption (EC).
```

SIMULATION TOOL AND EXPERIMENTAL SETUP

Implementation of this research work was carried out using CloudSim simulator. This section describes how the Simulation tool works, how it is setup, and what parameters it uses for execution.

About CloudSim

Calheiros et al. (2011) CloudSim is a platform for simulating the cloud computing environment. It is based on object-oriented Java programming language. It was developed by CLOUDS Laboratory at department of Computer Science and Engineering, University of Melbourne, Australia. Using CloudSim it is possible to simulate virtualization within data centers which allows for better experimentation and evaluation of cloud computing algorithms, meta heuristics, and protocols. Moreover, using CloudSim, users can simulate large-scale computing infrastructures and services, which include resource allocation, data centre, broker, allocation policies, scheduling etc.

Experimental Setup

The algorithms proposed in this research work are implemented in object-oriented Java Programming using Eclipse IDE and deployed to CloudSim toolkit version 4.0. The experiments were performed on a 64-bit operating system with a CPU (2.60GHz) and RAM (8 GB). A data centre with an x86 architecture and Linux OS was created. The characteristics of the host machine are set to CPU Capacity 1000 MIPS, RAM 4096 MB, and Disk space 2000000 MB. For each host, bandwidth is divided into two groups. Group-1 is a set of three host which takes values like {10,000, 15,000, and 20,000} and Group2 is a set of five host which takes values like {10000, 15000, 20000, 25000, 30000}. For VMs, number of CPU set to one and amount of bandwidth generated randomly between (5000, 10000), (5000, 15000) and (500, 20000).

Table 1. Comparison of TET, TEC, RT, and EC of Tabu Bayesian whale (TBW) algorithm with existing optimization algorithm (PSO, GA, PSO-GA, Whale) for LIGO workflow

Number of VMs	PSO				GA				PSO-GA				WHALE				TBW			
	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC
2	36.3	24.6	16.6	14.9	40.6	37.7	16.5	20.1	24.6	13.0	9.5	5.8	26.1	40.5	14.4	14.1	11.2	4.1	5.4	0.56
4	39.3	30.5	15.3	16.9	43.3	57.2	14.7	25.9	29.1	17.1	8.5	7.1	40.5	43.2	19.0	12.3	12.0	6.5	3.7	0.83
6	41.6	53.6	13.2	24.1	46.1	84.0	12.9	33.4	31.3	19.6	8.0	7.8	66.8	46.4	26.7	10.9	12.5	7.1	3.6	0.85
8	44.6	83.1	11.1	32.3	49.6	107.7	11.8	39.4	33.5	30.8	5.4	11.8	95.4	50.1	34.4	10.1	13.0	11.1	2.3	1.2
10	47.7	110.7	9.9	39.5	53.5	124.4	11.4	42.8	36.1	86.3	2.1	28.7	116.2	54.5	39.3	9.9	15.0	30.0	1.0	3.1
12	52.0	124.7	9.9	42.1	58.4	135.0	11.4	43.4	42.7	113.7	1.9	33.4	127.4	59.0	40.7	10.0	16.7	42.3	0.79	4.1
14	56.4	133.2	9.9	42.4	62.7	142.0	11.5	42.6	42.7	113.7	1.9	33.4	136.3	63.2	40.3	10.0	17.0	44.3	0.77	3.9
16	62.3	142.5	10.0	41.3	66.7	147.5	11.6	41.3	49.1	128.1	1.9	36.5	142.6	65.9	39.0	10.1	17.5	50.0	0.70	4.0
18	64.9	145.7	10.1	39.7	68.15	149.3	11.6	40.5	53.4	133.6	2.0	33.2	146.3	67.5	37.5	10.1	20.1	52.3	0.76	4.0
20	68.4	149.9	10.1	38.3	69.9	151.4	11.6	39.8	60.4	141.9	2.1	30.3	148.4	68.4	36.8	10.1	23.4	55.0	0.85	4.0

RESULTS AND DISCUSSION

The validation is performed via simulation-based experiments using cloudsim. The efficiency of TBW has been analysed by scheduling different workflow tasks on various VMs using TBW algorithm. The whole agenda of proposed study and its implementation is to reduce total cost and time consumed in whole process of mappings input tasks to VMs. Using cloud simulator, TBW (Tabu-Bayesian-Whale) scheduling algorithm has provided us better results as compared to existing scheduling and optimization algorithms. Table 1 shows the comparison of TET, TEC, RT, and EC of PSO, GA, PSO-GA, WHALE, and TBW which were implemented in various phases of this research. Here we have worked upon five different types of scientific workflows MONTAGE, CYBERSHAKE, LIGO, GENOME, and SIPHT. Figures 6 10 depicts the graphical comparison of TET, TEC, RT, and EC of the TBW algorithm with existing optimization algorithms using different number of active virtual machines.

As per the data in Table-1 we observe that TET, TEC, RT and EC are optimized in terms of TBW for LIGO, SIPHT, CYBERSHAKE, MONTAGE and GNOME scientific workflow. Figures 7-11 depict the graphical comparison of TET, TEC, RT, and EC of the TBW algorithm with existing optimization algorithms using the different number of active virtual machines. Figure 7 depicts the simulation results scheduling LIGO workflow, Figure 8 for CYBERSHAKE, Figure 9 for GENOME, Figure 10 for SIPHT and Figure 11 for MONTAGE workflow. X-axis of all the graphs shown in Figures 7-11 represent the number of active VMs used. Y-axis of graphs represents the time (in ms)

Figure 7. Simulation results of TET, TEC, RT, and EC parameters of scheduling LIGO workflow for different optimization algorithms

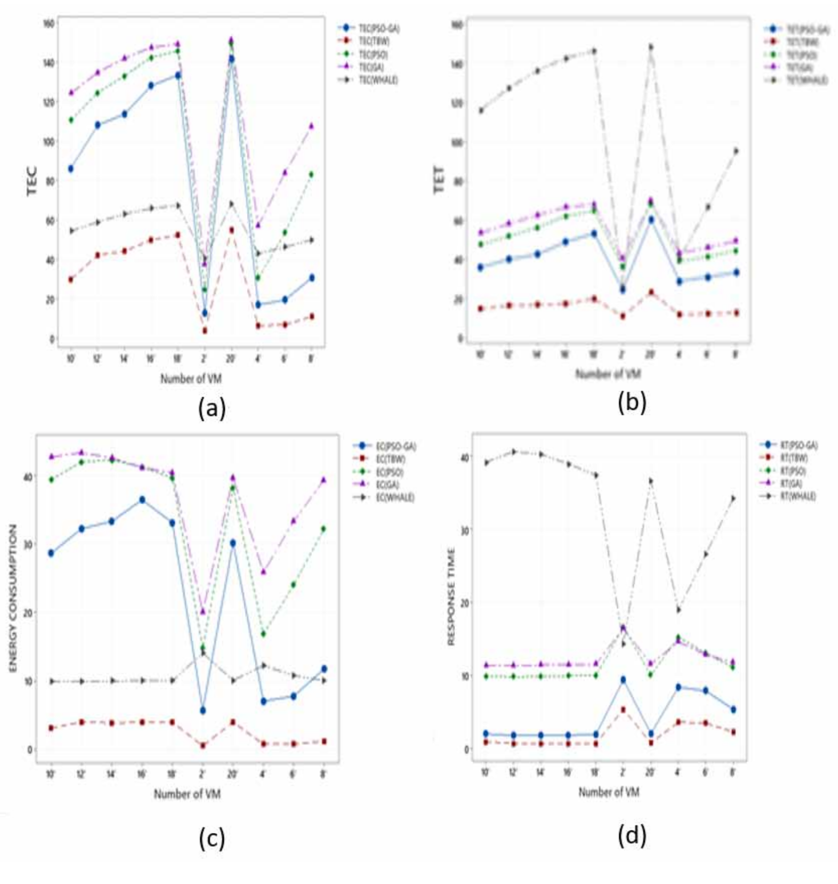


Table 2. Comparison of TET, TEC, RT, and EC of Tabu Bayesian whale (TBW) algorithm with existing optimization algorithm (PSO, GA, PSO-GA, whale) for CYBERSHAKE workflow

Number of VMs	PSO				GA				PSO-GA				WHALE				TBW			
	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC
2	51.7	35.2	17.5	21.5	57.1	58.3	16.0	23.0	39.8	15.4	12.9	13.3	43.3	57.7	20.0	13.5	5.8	11.8	2.4	1.1
4	55.6	54.5	14.3	21.5	61.4	81.2	13.9	22.8	44.0	21.8	10.1	13.4	66.1	62.3	19.9	11.8	8.1	12.4	3.3	1.1
6	59.5	80.7	11.9	21.4	66.2	103.1	12.7	22.5	47.3	44.4	5.3	13.9	90.8	67.6	19.6	11.0	15.9	13.4	5.9	1.1
8	64.5	103.9	11.0	20.9	72.0	119.8	12.4	22.2	51.5	73.3	3.5	13.2	110.6	73.6	19.3	10.8	27.7	14.8	9.3	1.1
10	70.0	120.3	10.8	20.7	78.2	131.0	12.3	22.1	55.9	100.4	2.8	12.9	123.9	79.8	19.1	10.9	38.8	16.2	12.0	1.2
12	76.9	130.6	10.8	20.6	84.4	138.5	12.4	22.1	62.0	114.0	62.0	114.0	132.6	85.2	19.1	10.9	45.4	17.0	13.4	1.1
14	83.0	137.5	10.9	20.6	89.0	143.2	12.5	22.1	68.2	122.3	68.2	122.3	138.8	89.2	19.1	11.0	48.8	18.1	13.5	1.1
16	88.7	143.0	11.0	20.6	92.4	146.3	12.5	22.1	76.6	131.5	76.6	131.5	142.3	91.4	19.1	11.0	52.3	20.2	12.9	1.1
18	90.7	144.7	11.0	20.6	93.5	147.2	12.5	22.1	80.3	134.7	80.3	134.7	144.2	92.6	19.1	11.1	53.6	21.7	12.4	1.2
20	93.2	146.8	11.1	20.6	94.7	148.3	12.6	22.1	85.2	138.8	85.2	138.8	145.3	93.2	19.1	11.1	54.9	23.3	11.8	1.3

Figure 8. Simulation results of TET, TEC, RT, and EC parameters of scheduling CYBERSHAKE workflow for different optimization algorithms

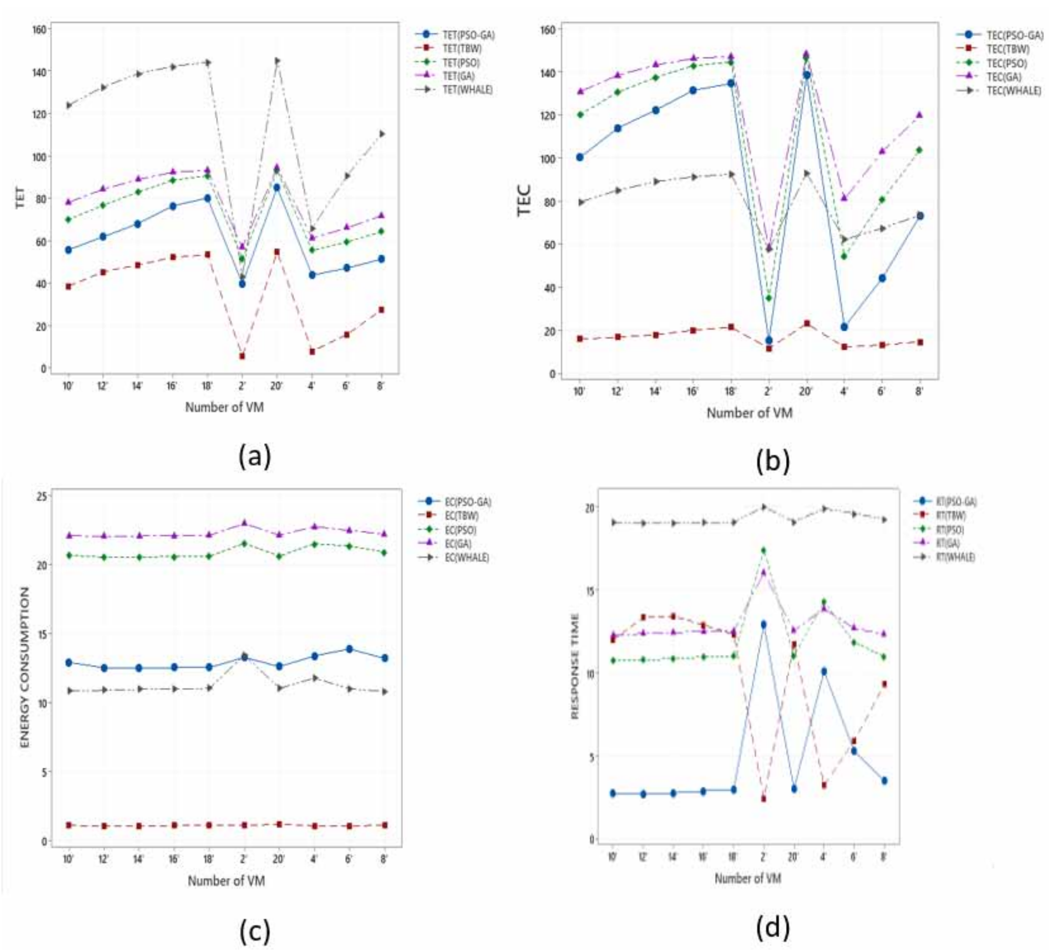


Table 3. Comparison of TET, TEC, RT, and EC of Tabu Bayesian whale (TBW) algorithm with existing optimization algorithm (PSO, GA, PSO-GA, whale) for GENOME workflow

Number of VMs	PSO				GA				PSO-GA				WHALE				TBW			
	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC
2	50.6	32.2	18.0	16.5	56.0	44.7	17.0	20.9	39.1	13.6	14.4	5.3	36.3	56.7	16.6	14.7	12.8	10.2	2.5	0.4
4	54.3	43.4	15.0	19.6	60.4	54.8	15.4	23.3	42.6	23.0	9.2	8.4	47.3	61.4	19.4	13.6	13.8	17.5	1.6	0.6
6	58.6	54.1	13.6	22.2	65.3	62.9	14.8	24.7	46.1	35.9	6.4	11.9	56.7	66.5	21.4	13.2	15.0	27.7	1.1	0.9
8	63.7	62.3	13.1	23.6	70.7	68.7	14.7	25.2	50.4	47.3	5.3	14.6	63.5	71.7	22.3	13.2	16.2	37.5	0.9	1.2
10	69.2	67.8	13.1	23.9	75.7	72.5	14.7	25.1	55.3	55.3	5.0	16.0	68.1	76.3	22.3	13.3	17.3	44.6	0.8	1.3
12	74.7	71.5	13.2	23.7	79.9	75.0	14.9	24.7	61.4	60.4	5.1	16.2	71.1	79.7	21.9	13.4	18.7	49.1	0.8	1.3
14	78.7	73.8	13.4	23.2	82.5	76.4	15.0	24.4	66.8	63.7	5.2	15.7	73.0	81.9	21.5	13.5	20.2	51.8	0.8	1.3
16	81.7	75.3	13.5	22.8	84.2	77.3	15.0	24.1	71.8	66.4	5.4	15.1	74.0	83.0	21.2	13.5	22.0	53.8	0.8	1.3
18	82.7	75.7	13.5	22.6	84.7	77.5	15.0	24.1	73.6	67.2	5.5	14.8	74.5	83.5	21.1	13.5	22.7	54.5	0.8	1.3
20	83.8	76.2	13.6	22.5	85.3	77.7	15.1	24.0	75.8	68.2	5.6	14.5	74.7	83.8	21.0	13.6	23.6	55.1	0.9	1.3

Figure 9. Simulation results of TET, TEC, RT, and EC parameters of scheduling GENOME workflow for different optimization algorithm

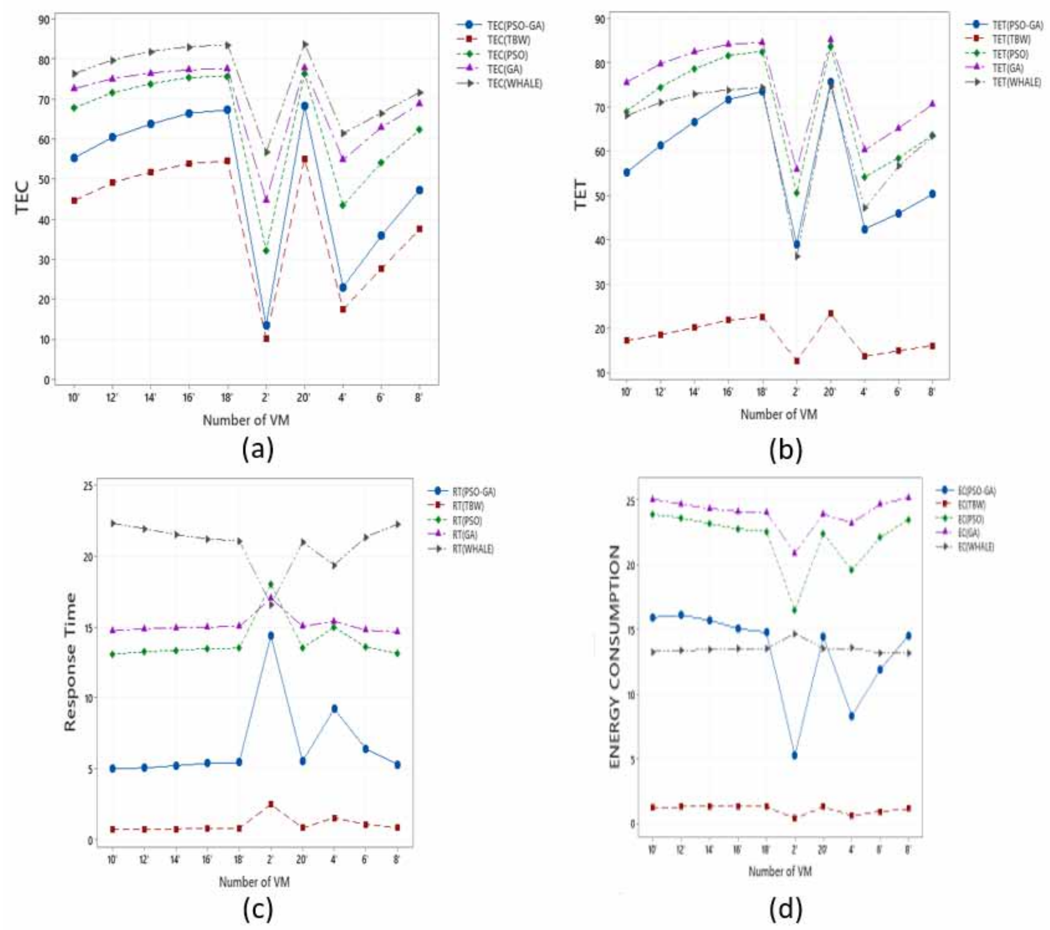


Table 4. Comparison of TET, TEC, RT, and EC of Tabu Bayesian whale (TBW) algorithm with existing optimization algorithm (PSO, GA, PSO-GA, whale) for SIPHT workflow

Number of VMs	PSO				GA				PSO-GA				WHALE				TBW			
	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC
2	41.0	64.6	10.9	27.2	45.5	71.4	12.4	28.9	30.3	52.1	2.9	19.1	65.6	45.7	25.7	10.9	13.6	15.9	1.7	0.8
4	43.9	69.6	10.9	27.3	48.9	77.3	12.4	29.2	32.9	56.5	2.9	19.1	71.1	49.3	26.0	10.9	14.7	25.8	1.1	1.2
6	47.3	75.5	10.9	27.7	52.7	83.7	12.4	29.5	35.7	61.4	2.9	19.2	77.3	53.0	26.4	10.9	16.0	37.0	0.9	1.6
8	51.1	82.1	10.9	28.1	56.4	90.2	12.4	29.8	39.1	67.1	2.9	19.6	84.0	56.6	26.7	10.9	17.1	46.5	0.7	1.8
10	55.1	88.9	10.9	28.4	59.8	95.9	12.4	29.8	43.1	74.0	2.9	20.2	90.4	59.5	26.9	10.9	18.3	53.0	0.7	1.9
12	58.6	94.9	10.9	28.4	62.3	100.3	12.4	29.8	47.2	81.2	2.9	20.5	95.6	61.6	26.8	10.9	19.8	57.0	0.7	1.9
14	61.1	99.3	10.9	28.3	63.9	103.0	12.4	29.7	51.0	87.7	2.9	20.4	99.2	62.8	26.7	10.9	21.4	59.7	0.7	2.0
16	62.7	102.0	10.9	28.1	64.8	104.5	12.4	29.6	53.5	91.9	2.9	20.3	101.1	63.4	26.6	10.9	22.7	61.2	0.7	2.0
18	63.3	103.1	10.9	28.1	65.0	104.9	12.4	29.6	54.9	94.4	2.9	20.1	101.9	63.6	26.6	10.9	23.5	62.0	0.8	2.0
20	63.7	103.8	10.9	28.0	65.2	105.3	12.4	29.5	55.7	95.8	2.9	20.0	102.3	63.7	26.5	10.9	23.9	62.3	0.8	2.0

Figure 10. Simulation results of TET, TEC, RT, and EC parameters of scheduling SIPHT workflow for different optimization algorithms

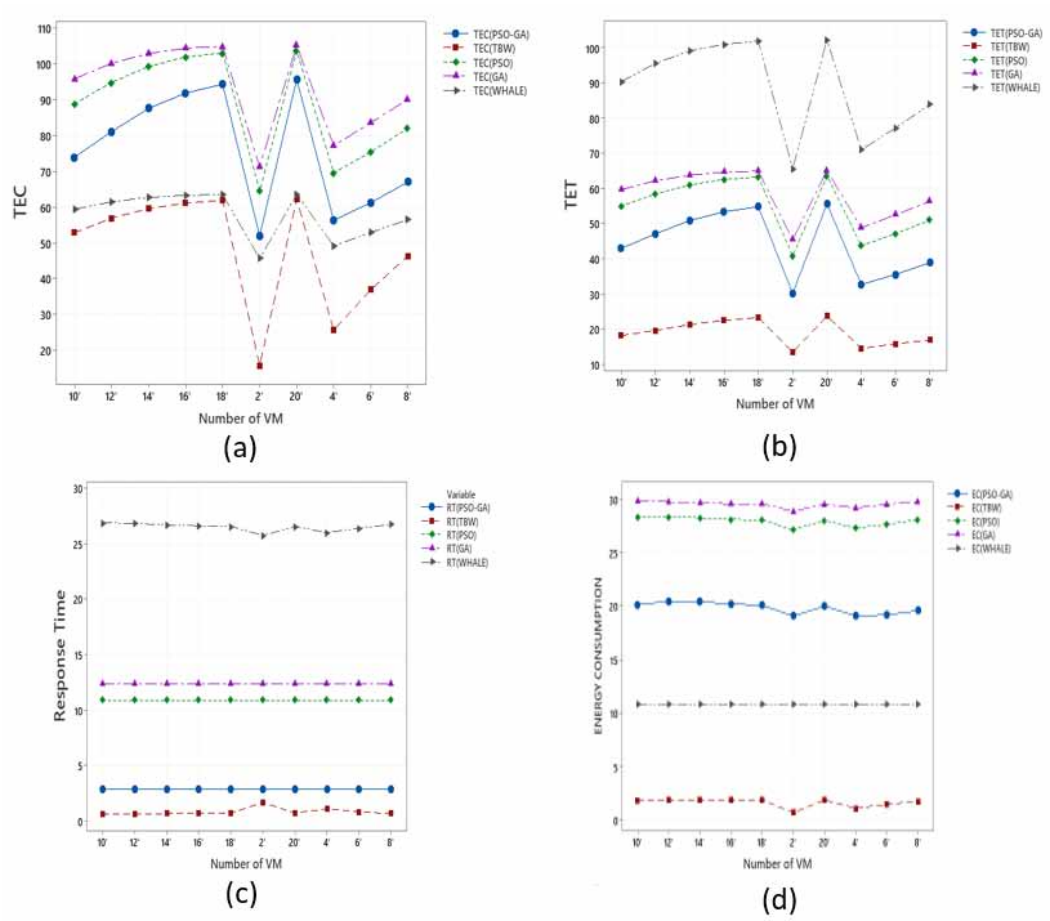
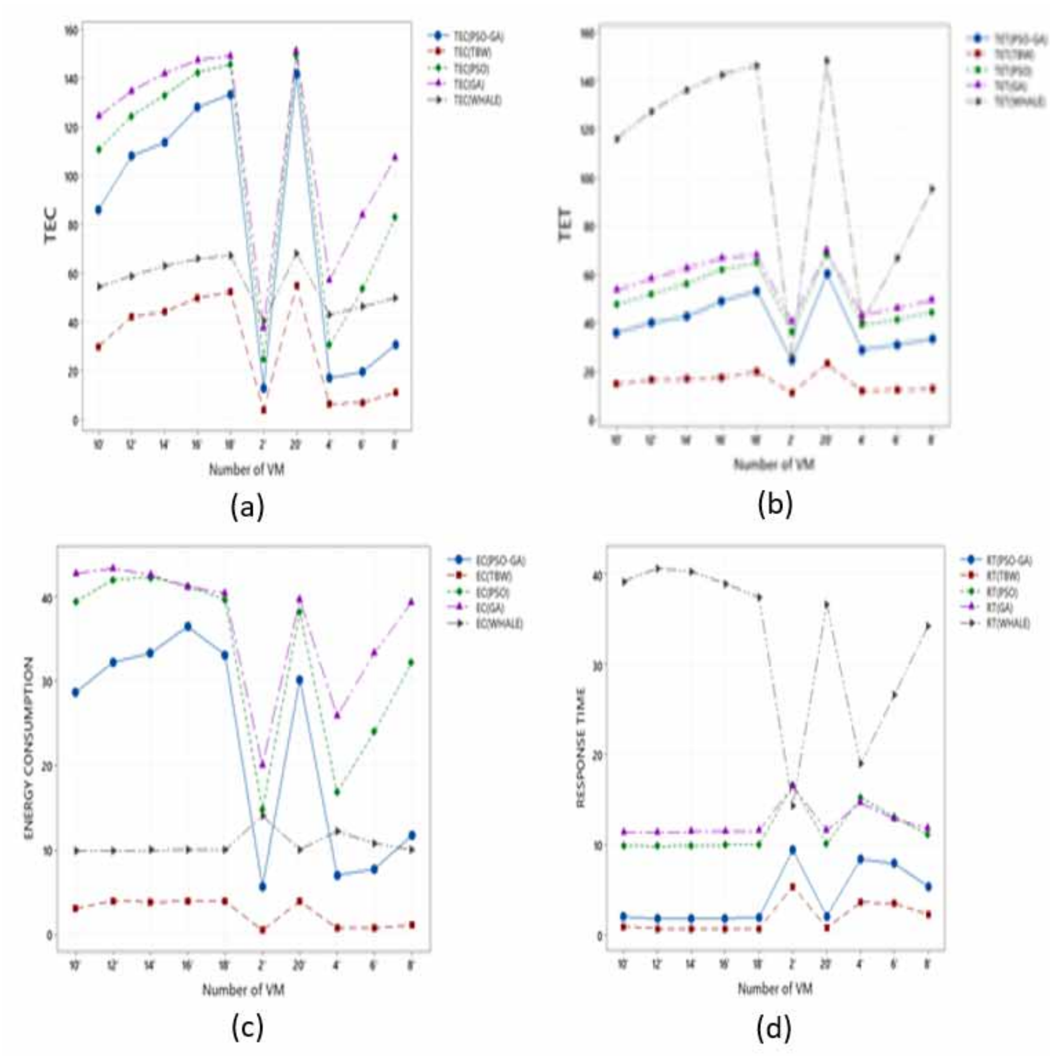


Table 5. Comparison of TET, TEC, RT, and EC of Tabu Bayesian whale (TBW) algorithm with existing optimization algorithm (PSO, GA, PSO-GA, whale) for MONTAGE workflow

Number of VMs	PSO				GA				PSO-GA				WHALE				TBW			
	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC	TET	TEC	RT	EC
2	38.4	63.1	10.8	32.3	43.2	70.3	12.3	33.7	27.6	50.3	2.7	24.7	64.4	43.5	30.7	10.8	10.2	16.8	1.8	1.3
4	41.6	68.6	10.8	32.2	46.8	76.5	12.3	33.7	30.3	54.9	2.8	24.2	70.2	47.2	30.7	10.8	11.4	27.4	1.2	2.0
6	45.2	74.8	10.8	32.2	50.5	82.9	12.3	33.7	33.4	60.2	2.8	24.0	76.7	50.8	30.8	10.8	12.5	37.8	1.0	2.5
8	49.1	81.5	10.8	32.3	54.1	89.0	12.3	33.6	37.1	66.6	2.8	24.3	83.2	54.1	30.7	10.8	13.7	45.7	0.9	2.8
10	52.8	87.9	10.8	32.1	57.0	94.1	12.3	33.3	41.1	73.6	2.8	24.4	89.0	56.6	30.4	10.8	15.1	51.0	0.9	2.9
12	55.9	93.1	10.8	31.8	59.1	97.7	12.3	33.0	45.1	80.4	2.8	24.2	93.4	58.2	30.1	10.8	16.6	54.4	0.9	2.9
14	57.9	96.7	10.8	31.4	60.3	99.8	12.3	32.7	48.2	85.8	2.8	23.7	96.2	59.1	29.8	10.8	18.1	56.5	1.0	2.9
16	59.1	98.7	10.8	31.2	60.9	100.8	12.3	32.6	50.2	89.2	2.8	23.4	97.6	59.5	29.6	10.8	19.1	57.6	1.0	2.9
18	59.5	99.4	10.8	31.1	61.1	101.1	12.3	32.6	51.3	91.1	2.8	23.1	98.1	59.7	29.6	10.8	19.7	58.1	1.0	2.9
20	59.7	99.8	10.8	31.1	61.2	101.3	12.3	32.6	51.7	91.8	2.8	23.1	98.3	59.7	29.6	10.8	19.9	58.3	1.0	2.9

Figure 11. Simulation results of TET, TEC, RT, and EC parameters of scheduling MONTAGE workflow for different optimization algorithm



for TET and RT parameter, Cost (in rupees) for TEC parameter, and Energy Consumption (in KWh) for EC parameter.

CONCLUSION

Scheduling complex workflows is a challenge in the field of Cloud- Computing Environment. There is a need to optimize the Cloud-based resources while scheduling the workflow. The scheduling of scientific workflows needs to be managed carefully in the virtualized infrastructure to optimize execution time, cost, and energy consumption. In this paper, a comprehensive scientific workflow scheduling framework named TBW is designed and implemented in a simulated environment to optimize these parameters. The proposed framework indulges different time and cost-related attributes in order to optimize overall execution time and cost for scheduling scientific workflow. Moreover, the phased architecture of the framework is designed to perform various predefined tasks in a sequential and synchronized manner. In order to achieve the overall effectiveness, the combination of three optimization approaches named Tabu Search, Bayesian Optimization, and Whale Optimization techniques is used.

In this research work, input workflow tasks are not randomly mapped to virtual machines but are first ranked using the distributed HEFT method and then scheduled on cloud-based machines. Afterwards, the optimization process starts which executes the TBW method to control time and cost parameters. We have incorporated a better optimization approach named TBW optimizer scheduler for optimizing the Total execution time, Total execution cost, Response time and Energy consumption parameters. The proposed approach has been implemented for MONTAGE, CYBERSHAKE, LIGO, GENOME, and SIPHT scientific workflows. The evaluated results have provided better results than the Whale optimization, GAPSO, GA and PSO approach for optimization. The proposed system is more effective as it has used effective optimization in a better way.

REFERENCES

Abrishami, S., Naghibzadeh, M., & Epema, D. H. (2013). Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds. *Future Generation Computer Systems*, *29*(1), 158–169. doi:10.1016/j.future.2012.05.004

Al-Maamari, A., & Omara, F. A. (2015). Task scheduling using pso algorithm in cloud computing environments. *International Journal of Grid and Distributed Computing*, *8*(5), 245–256. doi:10.14257/ijgdc.2015.8.5.24

Al-Qerem, A., Alauthman, M., Almomani, A., & Gupta, B. B. (2020). IoT transaction processing through cooperative concurrency control on fog–cloud computing environment. *Soft Computing*, *24*(8), 5695–5711. doi:10.1007/s00500-019-04220-y

Alakbarov, R. (2022). An Optimization Model for Task Scheduling in Mobile Cloud Computing. *International Journal of Cloud Applications and Computing*, *12*(1), 1–17. doi:10.4018/IJCAC.297102

Alkhanak, E. N., & Lee, S. P. (2018). A hyper-heuristic cost optimisation approach for scientific workflow scheduling in cloud computing. *Future Generation Computer Systems*, *86*, 480–506. doi:10.1016/j.future.2018.03.055

Alkhanak, E. N., Lee, S. P., Rezaei, R., & Parizi, R. M. (2016). Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues. *Journal of Systems and Software*, *113*, 1–26. doi:10.1016/j.jss.2015.11.023

Arabnejad, H., & Barbosa, J. G. (2014). A budget constrained scheduling algorithm for workflow applications. *Journal of Grid Computing*, *12*(4), 665–679. doi:10.1007/s10723-014-9294-7

Arabnejad, H., Barbosa, J. G., & Prodan, R. (2016). Lowtime complexity budget–deadline constrained workflow scheduling on heterogeneous resources. *Future Generation Computer Systems*, *55*, 29–40. doi:10.1016/j.future.2015.07.021

Arisdakessian, S., Wahab, O. A., Mourad, A., Otrok, H., & Kara, N. (2020). FoGMatch: An intelligent multi-criteria IoT-Fog scheduling approach using game theory. *IEEE/ACM Transactions on Networking*, *28*(4), 1779–1789. doi:10.1109/TNET.2020.2994015

Barker, A., & Van Hemert, J. (2007). Scientific workflow: a survey and research directions. In *International Conference on Parallel Processing and Applied Mathematics* (pp. 746–753). Springer.

Bhardwaj, A., & Kaushik, K. (2022). Predictive analytics-based cybersecurity framework for cloud infrastructure. *International Journal of Cloud Applications and Computing*, *12*(1), 1–20. doi:10.4018/IJCAC.297106

Bhushan, K., & Gupta, B. B. (2017). Security challenges in cloud computing: State-of-art. *International Journal of Big Data Intelligence*, *4*(2), 81–107. doi:10.1504/IJBDI.2017.083116

Bisht, J., & Vampugani, V. S. (2022). Load and cost-aware min-min workflow scheduling algorithm for heterogeneous resources in fog, cloud, and edge scenarios. *International Journal of Cloud Applications and Computing*, *12*(1), 1–20. doi:10.4018/IJCAC.2022010105

Byun, E.-K., Kee, Y.-S., Kim, J.-S., & Maeng, S. (2011). Cost optimized provisioning of elastic resources for application workflows. *Future Generation Computer Systems*, *27*(8), 1011–1026. doi:10.1016/j.future.2011.05.001

C., Nigam, G., Sharma, & Menghani. (2022). Comprehensive Review and Analysis on Mobile Cloud Computing and Users Offloading using Improved Optimization Approach for Edge Computing. *Int J Intell Syst Appl Eng, 10*(1).

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2011). Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software, Practice & Experience*, *41*(1), 23–50. doi:10.1002/spe.995

Casavant, T. L., & Kuhl, J. G. (1988). A taxonomy of scheduling in general-purpose distributed computing systems. *IEEE Transactions on Software Engineering*, *14*(2), 141–154. doi:10.1109/32.4634

Choudhary, A., Gupta, I., Singh, V., & Jana, P. K. (2018). A gsa based hybrid algorithm for bi-objective workflow scheduling in cloud computing. *Future Generation Computer Systems*, *83*, 14–26. doi:10.1016/j.future.2018.01.005

Deelman, E., Vahi, K., Juve, G., Rynge, M., Callaghan, S., Maechling, P. J., Mayani, R., Chen, W., Da Silva, R. F., & Livny, M. (2015). Pegasus, a workflow management system for science automation. *Future Generation Computer Systems*, *46*, 17–35. doi:10.1016/j.future.2014.10.008

Delavar, A. G., & Aryan, Y. (2014). Hsga: A hybrid heuristic algorithm for workflow scheduling in cloud systems. *Cluster Computing*, *17*(1), 129–137. doi:10.1007/s10586-013-0275-6

Dubey, K., Kumar, M., & Sharma, S. C. (2018). Modified HEFT Algorithm for Task Scheduling in Cloud Environment. *Procedia Computer Science*, *125*, 725–732. doi:10.1016/j.procs.2017.12.093

Elrotub, M., Bali, A., & Gherbi, A. (2021). Sharing VM resources with using prediction of future user requests for an efficient load balancing in cloud computing environment. *International Journal of Software Science and Computational Intelligence*, *13*(2), 37–64. doi:10.4018/IJSSCI.2021040103

Farhat, P., Sami, H., & Mourad, A. (2020). Reinforcement R-learning model for time scheduling of on-demand fog placement. *Journal of Supercomputing, 76*, 388-410.

Gaurav, A., Gupta, B. B., Alhalabi, W., Visvizi, A., & Asiri, Y. (2022). A comprehensive survey on DDoS attacks on various intelligent systems and it's defense techniques. *International Journal of Intelligent Systems*, *37*(12), 11407–11431. doi:10.1002/int.23048

Gaurav, A., Psannis, K., & Peraković, D. (2022). Security of cloud-based medical internet of things (miots): A survey. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–16. doi:10.4018/IJSSCI.285593

Ghose, M., Verma, P., Karmakar, S., & Sahu, A. (2017). Energy efficient scheduling of scientific workflows in cloud environment. In *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)* (pp. 170–177). IEEE. doi:10.1109/HPCC-SmartCity-DSS.2017.22

Gil, Y., Deelman, E., Ellisman, M., Fahringer, T., Fox, G., Gannon, D., Goble, C., Livny, M., Moreau, L., & Myers, J. (2007). Examining the challenges of scientific workflows. *Computer*, *40*(12), 24–32. doi:10.1109/MC.2007.421

Gupta, B. B., Joshi, R. C., & Misra, M. (2009). Defending against distributed denial of service attacks: issues and challenges. *Information Security Journal: A Global Perspective, 18*(5), 224-247.

Gupta, B. B., Li, K. C., Leung, V. C., Psannis, K. E., & Yamaguchi, S. (2021). Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system. *IEEE/CAA Journal of Automatica Sinica, 8*(12), 1877-1890.

Gupta, S., Gupta, B. B., & Chaudhary, P. (2018). Hunting for DOM-Based XSS vulnerabilities in mobile cloud-based online social network. *Future Generation Computer Systems*, *79*, 319–336. doi:10.1016/j.future.2017.05.038

Hammoud, A., Mourad, A., Otrok, H., Wahab, O. A., & Harmanani, H. (2020). Cloud federation formation using genetic and evolutionary game theoretical models. *Future Generation Computer Systems*, *104*, 92–104. doi:10.1016/j.future.2019.10.008

Hemrajani, N., Sharma, S., Upadhyay, A., Bhardwaj, Y., Kumar, A., & Sharma, G. (2022, June). Data management framework for IoT edge-cloud architecture for resource-constrained IoT application. *Journal of Discrete Mathematical Sciences & Cryptography*, *25*(4), 1093–1103. doi:10.1080/09720529.2022.2075086

Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66–73. doi:10.1038/scientificamerican0792-66

Huang, L., Chen, H., & Hu, T. (2013). Survey on resource allocation policy and job scheduling algorithms of cloud computing1. *Journal of Software*, *8*(2), 480–487. doi:10.4304/jsw.8.2.480-487

Jiang, J., Lin, Y., Xie, G., Fu, L., & Yang, J. (2017). Time and energy optimization algorithms for the static scheduling of multiple workflows in heterogeneous computing system. *Journal of Grid Computing*, *15*(4), 435–456. doi:10.1007/s10723-017-9391-5

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm. *Journal of Global Optimization*, *39*(3), 459–471. doi:10.1007/s10898-007-9149-x

Kouatli, I. (2020). The use of fuzzy logic as augmentation to quantitative analysis to unleash knowledge of participants' uncertainty when filling a survey: Case of cloud computing. *IEEE Transactions on Knowledge and Data Engineering*, *34*(3), 1489–1500. doi:10.1109/TKDE.2020.2993326

Kumar, A., Sharma, G., Jain, P., Upadhyay, A., Sharma, S., & Saini, A. K. (2022). Virtual environments testing in cloud service enviorment: A framework to optimize the performance of virtual applications. *International Journal of System Assurance Engineering and Management*, *13*(1), 1–15. doi:10.1007/s13198-021-01105-y

Kumar, A., & Sivakumar, P. (2022). Cat-squirrel optimization algorithm for VM migration in a cloud computing platform. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–23.

Kumar, S., Kumar, S., Ranjan, N., Tiwari, S., Kumar, T. R., Goyal, D., Sharma, G., Arya, V., & Rafsanjani, M. K. (2022). Digital watermarking-based cryptosystem for cloud resource provisioning. *International Journal of Cloud Applications and Computing*, *12*(1), 1–20. doi:10.4018/IJCAC.311033

Kuppusamy, P., Kumari, N. M. J., Alghamdi, W. Y., Alyami, H., Ramalingam, R., Javed, A. R., & Rashid, M. (2022). Job scheduling problem in fog-cloud-based environment using reinforced social spider optimization. *Journal of Cloud Computing (Heidelberg, Germany)*, *11*(1), 99. doi:10.1186/s13677-022-00380-9

Liang, Y., Liu, Y., & Gupta, B. B. (2022). PPRP: Preserving-privacy route planning scheme in VANETs. *ACM Transactions on Internet Technology*, *22*(4), 1–18. doi:10.1145/3430507

Liu, L., Zhang, M., Buyya, R., & Fan, Q. (2017). Deadline constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing. *Concurrency and Computation*, *29*(5), e3942. doi:10.1002/cpe.3942

Malawski, M., Juve, G., Deelman, E., & Nabrzyski, J. (2015). Algorithms for cost-and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds. *Future Generation Computer Systems*, *48*, 1–18. doi:10.1016/j.future.2015.01.004

Mehla, R., & Psannis, K. (2022). Multi-Authority Fine-Grained Data Sharing and Search Scheme for Cloud Banking Systems. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–18. doi:10.4018/IJSSCI.300360

Mehta, G., Juve, G., & Chen, W. (2016). *Workflow generator.* https://confluence. pegasus. isi. edu/display/pegasus/WorkflowGenerator

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51–67. doi:10.1016/j.advengsoft.2016.01.008

Mishra, A., Gupta, N., & Gupta, B. B. (2021). Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller. *Telecommunication Systems*, *77*(1), 47–62. doi:10.1007/s11235-020-00747-w

Nasr, A. A., El-Bahnasawy, N. A., & El-Sayed, A. (2014). Task scheduling optimization in heterogeneous distributed systems. *Int. J. Comput. Appl*, *107*(4), 5–12.

Negi, P., Mishra, A., & Gupta, B. B. (2013). *Enhanced CBF packet filtering method to detect DDoS attack in cloud computing environment*. arXiv preprint arXiv:1304.7073.

Onyebuchi, A., Matthew, U. O., Kazaure, J. S., Okafor, N. U., Okey, O. D., Okochi, P. I., ... Matthew, A. O. (2022). *Business demand for a cloud enterprise data warehouse in electronic healthcare computing: Issues and developments in e-healthcare cloud*. Academic Press.

Pietri, I., Malawski, M., Juve, G., Deelman, E., Nabrzyski, J., & Sakellariou, R. (2013). Energy-constrained provisioning for scientific workflow ensembles. In *2013 International Conference on Cloud and Green Computing* (pp. 34–41). IEEE. doi:10.1109/CGC.2013.14

Priya, S., & Kiranbir, K. (2018). Hybrid artificial bee colony and tabu search based power aware scheduling for cloud computing. *International Journal of Intelligent Systems and Applications*, *10*(7), 39–47. doi:10.5815/ijisa.2018.07.04

Quamara, M., Gupta, B. B., & Yamaguchi, S. (2019, October). MQTT-driven remote temperature monitoring system for IoT-based smart homes. In *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)* (pp. 968-970). IEEE. doi:10.1109/GCCE46687.2019.9015603

Rajput, R. K. S., Goyal, D., Pant, A., Sharma, G., Arya, V., & Rafsanjani, M. K. (2022). Cloud data centre energy utilization estimation: Simulation and modelling with idr. *International Journal of Cloud Applications and Computing*, *12*(1), 1–16. doi:10.4018/IJCAC.311035

Reddy, G. N., & Kumar, S. P. (2017). Multi objective task scheduling algorithm for cloud computing using whale optimization technique. In *International Conference on Next Generation Computing Technologies* (pp. 286–297). Springer.

Rimal, B. P., & Maier, M. (2016). Workflow scheduling in multi-tenant cloud computing environments. *IEEE Transactions on Parallel and Distributed Systems*, *28*(1), 290–304. doi:10.1109/TPDS.2016.2556668

Rodriguez, M. A., & Buyya, R. (2014). Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds. *IEEE Transactions on Cloud Computing*, *2*(2), 222–235. doi:10.1109/TCC.2014.2314655

Rodriguez, M. A., & Buyya, R. (2017). A taxonomy and survey on scheduling algorithms for scientific workflows in iaas cloud computing environments. *Concurrency and Computation*, *29*(8), e4041. doi:10.1002/cpe.4041

Sandhu, R., & Lakhwani, K. (2022). Enhanced scientific workflow scheduling in cloud system. In *ICCCE 2021* (pp. 133–139). Springer. doi:10.1007/978-981-16-7985-8_14

Shahzad, F., Mannan, A., Javed, A. R., Almadhor, A. S., Baker, T., & Al-Jumeily, O. B. E. (2022). Cloud-based multiclass anomaly detection and categorization using ensemble learning. *Journal of Cloud Computing (Heidelberg, Germany)*, *11*(1), 1–12. doi:10.1186/s13677-022-00329-y

Sharma, S., & Sajid, M. (2021). Integrated fog and cloud computing issues and challenges. *International Journal of Cloud Applications and Computing*, *11*(4), 174–193. doi:10.4018/IJCAC.2021100110

Shaw, S. B., & Singh, A. (2014). A survey on scheduling and load balancing techniques in cloud computing environment. In 2014 international conference on computer and communication technology (ICCCT) (pp. 87–95). IEEE. doi:10.1109/ICCCT.2014.7001474

Shi, Z., & Dongarra, J. J. (2006). Scheduling workflow applications on processors with different capabilities. *Future Generation Computer Systems*, *22*(6), 665–675. doi:10.1016/j.future.2005.11.002

Singh, A., & Kumar, R. (2021). A two-phase load balancing algorithm for cloud environment. *International Journal of Software Science and Computational Intelligence*, *13*(1), 38–55. doi:10.4018/IJSSCI.2021010103

Singh, R., & Singh, S. (2013). Score based deadline constrained workflow scheduling algorithm for cloud systems. *International Journal on Cloud Computing: Services and Architecture*, *3*(6), 31–41. doi:10.5121/ijccsa.2013.3603

Sorkhoh, I., Ebrahimi, D., Assi, C., Sharafeddine, S., & Khabbaz, M. (2020). An infrastructure-assisted workload scheduling for computational resources exploitation in the fog-enabled vehicular network. *IEEE Internet of Things Journal*, *7*(6), 5021–5032. doi:10.1109/JIOT.2020.2975496

Sossa, M. A. R. (2016). *Resource provisioning and scheduling algorithms for scientific workflows in cloud computing environments* [PhD thesis]. University of Melbourne, Department of Computing and Information Systems.

Stergiou, C. L., Psannis, K. E., & Gupta, B. B. (2021). InFeMo: Flexible big data management through a federated cloud system. *ACM Transactions on Internet Technology*, *22*(2), 1–22. doi:10.1145/3426972

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons. doi:10.1002/9780470496916

Vecchiola, C., Pandey, S., & Buyya, R. (2009). Highperformance cloud computing: A view of scientific applications. In *2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks* (pp. 4–16). IEEE.

Verma, A., & Kaushal, S. (2015). Cost-time efficient scheduling plan for executing workflows in the cloud. *Journal of Grid Computing*, *13*(4), 495–506. doi:10.1007/s10723-015-9344-9

Vöckler, J.-S., Juve, G., Deelman, E., Rynge, M., & Berriman, B. (2011). Experiences using cloud computing for a scientific workflow application. *Proceedings of the 2nd international workshop on Scientific cloud computing*, 15–24. doi:10.1145/1996109.1996114

Wahab, O. A., Bentahar, J., Otrok, H., & Mourad, A. (2017). Optimal load distribution for the detection of VM-based DDoS attacks in the cloud. *IEEE Transactions on Services Computing*, *13*(1), 114–129. doi:10.1109/TSC.2017.2694426

Xu, Z., He, D., Vijayakumar, P., Gupta, B. B., & Shen, J. (2021). Certificateless public auditing scheme with data privacy and dynamics in group user model of cloud-assisted medical wsns. *IEEE Journal of Biomedical and Health Informatics*. PMID:34788225

Yang, J., & Chen, Z. (2010). Cloud computing research and security issues. In *2010 International Conference on Computational Intelligence and Software Engineering* (pp. 1–3). IEEE.

Yu, J., & Buyya, R. (2005). A taxonomy of workflow management systems for grid computing. *Journal of Grid Computing*, *3*(3-4), 171–200. doi:10.1007/s10723-005-9010-8

Zhang, L., Li, K., Li, C., & Li, K. (2017). Bi-objective workflow scheduling of the energy consumption and reliability in heterogeneous computing systems. *Information Sciences*, *379*, 241–256. doi:10.1016/j.ins.2016.08.003

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*, *1*(1), 7–18. doi:10.1007/s13174-010-0007-6

Zhang, Z., Sun, R., Zhao, C., Wang, J., Chang, C. K., & Gupta, B. B. (2017). CyVOD: A novel trinity multimedia social network scheme. *Multimedia Tools and Applications*, *76*(18), 18513–18529. doi:10.1007/s11042-016-4162-z

Zhao, Y., Fei, X., Raicu, I., & Lu, S. (2011). Opportunities and challenges in running scientific workflows on the cloud. In *2011 International Conference on Cyber Enabled Distributed Computing and Knowledge Discovery* (pp. 455–462). IEEE. doi:10.1109/CyberC.2011.80

Zhu, Z., Zhang, G., Li, M., & Liu, X. (2015). Evolutionary multi-objective workflow scheduling in cloud. *IEEE Transactions on Parallel and Distributed Systems*, *27*(5), 1344–1357. doi:10.1109/TPDS.2015.2446459